

Sistemas de Inteligencia Artificial

Métodos de Búsqueda Informados y no Informados

Trabajo Práctico Especial N° 1

Sokoban

Grupo 8:

Federico Tedin (53048)

Ignacio Rivera (53029)

Javier Fraire (53023)

Objetivo

El objetivo del trabajo es crear un sistema de producción para resolver un juego conocido como Sokoban. Para ésto, se utilizarán métodos de búsqueda no informados, tales como BFS, y metodos de busqueda informados, como A* y Greedy search.

Problema

El juego a resolver es el Sokoban, éste cuenta con un tablero donde se haya un jugador, una cantidad de cajas que deben ser posicionadas en ciertos lugares de “gol” en el tablero. El objetivo del juego es ubicar todas las cajas en un casillero de “gol”. Para mover una caja, el jugador se debe poner adyacente a la misma y moverse en una dirección, luego ésta se desplazará un casillero en la misma dirección siempre y cuando no haya otra caja o una pared que impida el movimiento.

Algoritmo genérico de búsqueda

Se utilizó el algoritmo de búsqueda provisto por la cátedra que consiste en los siguientes pasos:

1. Crear un árbol de búsqueda, T, que sólo esté formado por el nodo n_0 . Insertar n_0 en la lista de nodos FRONTERA.
2. Crear la lista de nodos EXPLORADOS, inicialmente vacía.
3. Sí FRONTERA está vacía, el algoritmo termina con falla.
4. Extraer el primer nodo de FRONTERA e insertarlo en EXPLORADOS. Llamar a éste nodo n.
5. Si n es el nodo objetivo, salir del algoritmo y devolver la solución, que estará formada por el camino definido por los arcos que llevan de n a n_0 en el árbol T.
6. Expandir el nodo n y generar el conjunto de nodos sucesores M. Incorporar los elementos de M como sucesores de n en el árbol T, creando los arcos correspondientes entre n y cada uno de los nodos de M.
7. Reordenar FRONTERA de acuerdo con algún esquema arbitrario o de acuerdo con una heurística determinada.
8. Ir al paso 3.

Reglas Aplicables

Para la resolución del problema se identificaron y aplicaron 4 reglas. Todas se refieren a los distintos movimientos que puede realizar el jugador, estas reglas son:

- Mover el jugador un casillero en dirección ($x = 1, y = 0$) (RIGHT/DERECHA)
- Mover el jugador un casillero en dirección ($x = -1, y = 0$) (LEFT/IZQUIERDA)
- Mover el jugador un casillero en dirección ($x = 0, y = -1$) (UP/ARRIBA)
- Mover el jugador un casillero en dirección ($x = 0, y = 1$) (DOWN/ABAJO)

Tiempos de Prueba

Para todas las pruebas que se hicieron para este trabajo, se tomó como cota superior temporal los 10 minutos; si el algoritmo no encuentra una solución al problema en este tiempo se lo categoriza en las tablas como SR (sin respuesta).

Breadth First

El primer método de búsqueda no informada que se implementó es el Breadth First Search (BFS). Éste método analiza todos los nodos de un nivel de profundidad del árbol de búsqueda antes de pasar a analizar el proximo nivel. Pueden verse los resultados en la tabla N° 1 del anexo.

Depth First

El segundo método de búsqueda no informada que se implementó es el Depth First Search, que se encarga de buscar en profundidad, a diferencia del método previamente mencionado que busca a lo ancho del árbol de búsqueda. Se pueden ver los resultados en la tabla N °2 del anexo.

Profundización iterativa

El último método de búsqueda no informada que se implementó es el método de profundización iterativa. Este consiste en realizar DFS con un nivel como cota superior empezando en 1 y terminando cuando se haya un nodo objetivo. En cada iteración del algoritmo, se aumenta el nivel de búsqueda. Los resultados de aplicar este algoritmo se hallan en la tabla N° 3 del anexo.

Heurística N° 1

La primer heurística se genera utilizando la suma de las distancias directas de las cajas a los casilleros de gol no ocupados. Esta heurística cuenta con una variante, agregar a la suma mencionada un factor al que llamamos '*deadlock*'. Este le agrega el número máximo de un entero Java al valor de la heurística en el caso de que una caja se encuentre en una posición de la cual no se pueda mover, por ejemplo las esquinas del mapa, de esta manera removiendo rápidamente los casos en los que el juego se vuelve irresoluble.

Ambas variantes de esta heurística no son admisibles, porque se sobreestima el costo cuando se suma la distancia a todos los casilleros de gol libres.

Heurística N° 2

La segunda heurística implementada utiliza la distancia manhattan de las distintas cajas con respecto a todos los casilleros de gol que no se encuentran ocupados. Antes de hacer el cálculo de las distancias Manhattan, se verifica que no haya ninguna caja encerrada

por una esquina de paredes (deadlock) en cuyo caso la heurística retorna el máximo entero posible.

De manera similar a la anterior, ésta heurística no es admisible ya que al sumar la distancia de todas las cajas a cada objetivo, se está sobreestimando la distancia total que necesitaría ser recorrida por el jugador para terminar el mapa.

Diferencias

La gran diferencia entre las dos heurísticas es que la primera utiliza la distancia en línea recta desde una caja a los distintos casilleros de gol mientras que la segunda utiliza la distancia Manhattan, que es la suma de las diferencias en las distancias en las coordenadas x e y. Además, la segunda heurística suma las distancias, por cada casillero gol no ocupado, a todas las cajas *que no estén sobre otro casillero gol*, mientras que la primera suma hacia todas las cajas. Este cambio menor permite que los valores devueltos por la segunda heurística reflejen mejor el estado del mapa.

Greedy search

Uno de los métodos de búsqueda informado que se implementaron es el Greedy search. Éste se caracteriza por elegir siempre el nodo donde el valor de la heurística sea menor. Se probó utilizando las heurísticas previamente mencionadas sobre los 6 mapas y los resultados pueden encontrarse en las tablas de 4 a la 6 en el anexo.

Como podemos ver en las tablas se encuentra un resultado para todos los mapas excepto el 6 tanto para la variante con deadlock activado como la variante sin el deadlock. Cabe destacar que la variante sin el deadlock es más rápida para los mapas 2 y 3 que no cuentan con paredes dentro del tablero mientras que la otra variante es más rápida para el caso contrario. Ésto se debe a que la segunda descarta los casos en los que las cajas se encuentran bloqueados dentro del tablero por paredes u otras cajas de una manera más veloz.

En la tercer tabla están los resultados de utilizar este metodo de busqueda con la segunda heurística, donde se ve como es mejor que las anteriores en los mapas 1 y 4, mientras que en los otros 3 resulta peor. De la misma manera no encuentra una solución para el mapa 6.

A*

El otro método de búsqueda informada que se implementó es el método A*. Éste se caracteriza por calcular la función dada por la suma del resultado de la funcion heuristica del nodo más el costo de dicho nodo. De la misma manera que con el Greedy, se probó utilizando las heurísticas previamente desarrolladas y los resultados fueron los siguientes:

Los resultados se pueden encontrar en las tablas 7 a 9 en el anexo. Viendo la tabla donde se encuentran los resultados de aplicar la primer heurística sin el deadlock podemos ver claramente que es mucho más eficiente que el método Greedy, ya que éste logra encontrar una solución para el mapa 6 en menos de 4 segundos. También las soluciones de los otros mapas son encontradas mucho más rápido que con el método anterior. Es importante destacar que como puede verse en la tabla donde están los resultados de aplicar la primer heurística en su variante con deadlock, no se puede encontrar resultado para 3 mapas (1, 5 y 6). Aun así las soluciones para los otros 3 mapas restantes se encuentran también considerablemente más rápido que utilizando el método Greedy.

En la última tabla podemos ver los resultados de utilizar éste método de búsqueda con la segunda heurística. No termina de resolver en el tiempo máximo estipulado el mapa 5 y el mapa 1 lo resuelve pero muy cerca del límite de tiempo. Al mismo tiempo, obtiene resultados buenos para los otros 4 mapas.

Conclusiones

Como se puede observar, los métodos de búsqueda no informados no son una buena forma de resolver éste problema ya que el árbol de búsqueda es de grandes dimensiones, y al no hacer ningún tipo de poda ni utilizar alguna heurística, las búsquedas, en su mayoría, exceden los 10 minutos impuestos como cota superior.

En lo que respecta a lo métodos informados, basándonos en los resultados vemos como en este caso la mejor forma de resolver el problema es utilizando A* con la primer heurística propuesta. A diferencia de Greedy, A* logró terminar todos los mapas antes del tiempo límite. También podemos apreciar como los tiempos de resolución son en su mayoría considerablemente más cortos, dentro de las dimensiones en los que se los está midiendo.

Si se hace una comparación entre las heurísticas propuestas, en lo que respecta a Greedy, hay casos en la que la primera es considerablemente mejor y otras en que la segunda tiene la ventaja. Ésto no se observa en el método de búsqueda A*, claramente la mejor opción a la hora de resolver es usar la primer heurística en su variante sin deadlock donde los tiempos son sumamente más cortos que el utilizado de la heurística que suma las distancias Manhattan.

ANEXO

Nodos Ex. = Nodos Expandidos

Nodos F. = Nodos Frontera

Nodos T. = Nodos Totales

Nivel P. = Nivel de profundidad

Tabla N° 1 (Breadth First):

	Mapa 1	Mapa 2	Mapa 3	Mapa 4	Mapa 5	Mapa 6
Tiempo (ms)	SR	47	SR	432000	SR	SR
Nodos Ex (nodos)	SR	80	SR	24696	SR	SR
Nodos F. (nodos)	SR	16	SR	8468	SR	SR
Nodos T. (nodos)	SR	96	SR	33164	SR	SR
Nivel P.	SR	9	SR	22	SR	SR

Tabla N° 2 (Depth First):

	Mapa 1	Mapa 2	Mapa 3	Mapa 4	Mapa 5	Mapa 6
Tiempo (ms)	SR	63	SR	SR	SR	SR
Nodos Ex (nodos)	SR	129	SR	SR	SR	SR
Nodos F. (nodos)	SR	12	SR	SR	SR	SR
Nodos T. (nodos)	SR	141	SR	SR	SR	SR
Nivel P. (nodos)	SR	17	SR	SR	SR	SR

Tabla N° 3 (Iterativo):

	Mapa 1	Mapa 2	Mapa 3	Mapa 4	Mapa 5	Mapa 6
Tiempo (ms)	SR	102	SR	SR	SR	SR
Nodos Ex (nodos)	SR	165	SR	SR	SR	SR
Nodos F. (nodos)	SR	5	SR	SR	SR	SR
Nodos T. (nodos)	SR	170	SR	SR	SR	SR
Nivel P.	SR	9	SR	SR	SR	SR

Tabla N° 4 (Greedy con primer heurística sin deadlock):

	Mapa 1	Mapa 2	Mapa 3	Mapa 4	Mapa 5	Mapa 6
Tiempo (ms)	9650	26	388	11838	3439	SR
Nodos Ex (nodos)	3937	16	391	5250	2140	SR
Nodos F. (nodos)	235	8	519	263	368	SR
Nodos T. (nodos)	4173	24	138	5513	2508	SR
Nivel P.	98	11	88	160	135	SR

Tabla N° 5 (Greedy con primer heurística con deadlock):

	Mapa 1	Mapa 2	Mapa 3	Mapa 4	Mapa 5	Mapa 6
Tiempo (ms)	8083	20	456	69851	3851	SR
Nodos Ex (nodos)	3261	16	381	12328	2140	SR
Nodos F. (nodos)	211	8	138	487	368	SR
Nodos T. (nodos)	3472	24	519	12816	2508	SR
Nivel P.	98	11	88	210	135	SR

Tabla N° 6 (Greedy con segunda heurística):

	Mapa 1	Mapa 2	Mapa 3	Mapa 4	Mapa 5	Mapa 6
Tiempo (ms)	7970	23	515	4137	53915	SR
Nodos Ex (nodos)	3068	16	483	2673	8334	SR
Nodos F. (nodos)	168	8	157	222	1147	SR
Nodos T. (nodos)	3236	24	640	2895	9481	SR
Nivel P.	70	11	129	102	906	SR

Tabla N° 7 (A* con primer heurística sin deadlock):

	Mapa 1	Mapa 2	Mapa 3	Mapa 4	Mapa 5	Mapa 6
Tiempo (ms)	1410	40	322	145	619	3993
Nodos Ex (nodos)	896	257	288	121	405	1407
Nodos F. (nodos)	305	10	173	90	242	1025
Nodos T. (nodos)	1201	37	461	211	648	2432
Nivel P.	40	9	26	22	35	72

Tabla N° 8 (A* con primer heurística con deadlock):

	Mapa 1	Mapa 2	Mapa 3	Mapa 4	Mapa 5	Mapa 6
Tiempo (ms)	SR	43	312	237	SR	SR
Nodos Ex (nodos)	SR	68	288	160	SR	SR
Nodos F. (nodos)	SR	8	173	87	SR	SR
Nodos T. (nodos)	SR	76	461	247	SR	SR
Nivel P.	SR	9	26	22	SR	SR

Tabla N° 9 (A* con segunda heurística):

	Mapa 1	Mapa 2	Mapa 3	Mapa 4	Mapa 5	Mapa 6
Tiempo (ms)	250061	30	826	156	SR	6300
Nodos Ex (nodos)	22956	25	901	72	SR	2024
Nodos F. (nodos)	199	7	2834	69	SR	1319
Nodos T. (nodos)	23155	32	241184	141	SR	3343
Nivel P.	40	9	23	22	SR	88